JOURNAL OF COMPUTER AND SYSTEM SCIENCES 12, 352–367 (1976)

Properties of Code Events and Homomorphisms over Regular Events

K. HASHIGUCHI AND N. HONDA

Research Institute of Electrical Communication, Tohoku University, Sendai 980, Japan

Received May 30, 1974; revised February 7, 1975

An event of the form $(w_1 \cup \cdots \cup w_n)^*$, each w_i being a word, is called a code event by R. McNaughton and S. Papert. In this paper, first, conditions are obtained for a code event to be noncounting, locally testable, and strictly locally testable, respectively. Next it is shown that a homomorphism $f\colon \Sigma_1^* \to \Sigma_2^*$ preserves strict local testability iff f is injective and the code event $f(\Sigma_1^*)$ is strictly locally testable. Similar results are obtained for locally testable events and noncounting events.

1. INTRODUCTION

An event of the form $(w_1 \cup \cdots \cup w_n)^*$, each w_i being a word, is called a code event by McNaughton and Papert [1]. They presented conditions for a code event w^* to be locally testable and noncounting, respectively, "in the spirit of giving examples." They also introduced the notion of local parsability and presented some results on code events. And they raised the following questions. Let R be a code event. Q1. Under what conditions is R noncounting? Q2. Under what conditions is R locally testable? Q3. If R is unambiguous and locally testable, then is it locally parsable?

In this paper we first give answers to these questions, which include an affirmative answer to Q3, and then develop some properties of homomorphisms. Precisely, in Section 3 conditions for a code event to be noncounting and locally testable, respectively, are obtained, by utilizing the theorems on syntactic semigroups due to Schützenberger [3] and McNaughton and Zalcstein [2], respectively. It is also shown that a code event is strictly locally testable iff it is locally parsable. In Section 4, it is shown that a homomorphism $f\colon \Sigma_1^* \to \Sigma_2^*$ preserves strict local testability iff f is injective and the code event $f(\Sigma_1^*)$ is strictly locally testable, by utilizing the result in Section 3. Similar results are obtained for locally testable events and noncounting events.

2. PRELIMINARIES

We assume that the reader is familiar with regular expressions, regular events, and finite automata [4]. In this section we present notation and basic definitions after [1]. Let Σ be a finite nonempty alphabet; Σ^*, the set of all words over Σ; λ, the null word;

Σ^+, the set of all nonnull words over Σ; and $\varnothing$, the empty event. For $w \in \Sigma^*$, let $| w |$ denote the length of w and $\#Q$ the cardinality of the set Q. Let $A = \langle \Sigma, Q, \delta, S, F \rangle$ denote a (finite) automaton over an input alphabet Σ, where Q is the set of states, δ is the transition function $\delta: Q \times \Sigma \to Q$, $S \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. Regular expressions use the operators union ($\cup$), intersection ($\cap$), complement ($-$), concatenation ($\cdot$), and star ($*$). For $R_1, R_2 \subseteq \Sigma^*$, $R_1 - R_2$ denotes the set difference.

DEFINITION 2.1. An event $R \subseteq \Sigma^*$ is noncounting iff R is regular and there exists an integer $k \geqslant 0$ such that for all $x, y, z \in \Sigma^*$, $xy^kz \in R$ iff $xy^{k+1}z \in R$. Let NC denote the class of noncounting events.

DEFINITION 2.2. Let k be a positive integer. For $w \in \Sigma^+$ of length $\geqslant k$, let $L_k(w)$, $R_k(w)$, $I_k(w)$ be the prefix of w of length k, the suffix of w of length k, and the set of interior solid subwords of length k, respectively. If $| w | = k$ or $k + 1$, then $I_k(w) = \varnothing$. Let $T_k(w)$ be the k-test vector of w, $\langle L_k(w), I_k(w), R_k(w) \rangle$.

DEFINITION 2.3. An event $R \subseteq \Sigma^*$ is strictly k-testable iff there exist $\alpha, \beta, \gamma \subseteq \Sigma^k$ such that for all $w \in \Sigma^*$ of length $\geqslant k$, $w \in R$ iff $L_k(w) \in \alpha$, $I_k(w) \subseteq \beta$, and $R_k(w) \in \gamma$. An event is strictly locally testable iff it is strictly k-testable for some $k \geqslant 1$.

DEFINITION 2.4. An event $R \subseteq \Sigma^*$ is k-testable iff for all $w, w' \in \Sigma^*$ of length $\geqslant k$, if $T_k(w) = T_k(w')$, then $w \in R$ iff $w' \in R$. An event is locally testable iff it is k-testable for some $k \geqslant 1$.

One can easily see that the following definition is equivalent to the above original one in [1]. For $R \subseteq \Sigma^*$, let $L_k(R) = \{L_k(w) \mid | w | \geqslant k \text{ and } w \in R\}$, $R_k(R) = \{R_k(w) \mid | w | \geqslant k \text{ and } w \in R\}$, $I_k(R) = \{x \mid x \in I_k(w) \text{ for some } w \in R \text{ of length } \geqslant k\}$, $T_k(R) = \{T_k(w) \mid | w | \geqslant k \text{ and } w \in R\}$, and $V_k(R) = \{\langle x, \beta, y \rangle \mid x \in L_k(R), \beta \subseteq I_k(R) \text{ and } y \in R_k(R)\}$. An event $R \subseteq \Sigma^*$ is k-testable iff for all $w \in \Sigma^*$ of length $\geqslant k$, $w \in R$ iff $T_k(w) \in T_k(R)$. An event $R \subseteq \Sigma^*$ is strictly k-testable iff for all $w \in \Sigma^*$ of length $\geqslant k$ $w \in R$ iff $T_k(w) \in V_k(R)$.

Note that if $R \subseteq \Sigma^*$ is strictly k-testable, then it is k-testable and $T_k(R) = V_k(R) \cap T_k(\Sigma^*)$. Let LT (respectively, SLT) be the class of locally testable events (respectively, strictly locally testable events).

SLT is not closed under the Boolean operations. The closure of SLT under the Boolean operations is LT. LT is not closed under concatenation, and the closure of LT under the Boolean operations and concatenation is NC [1].

In this paper we assume that whenever a code event R is denoted by a regular expression $(w_1 \cup \cdots \cup w_n)^*$, the expression $(w_1 \cup \cdots \cup w_n)^*$ is reduced in the following sense. For each i $(1 \leqslant i \leqslant n)$, $w_i \notin ((w_1 \cup \cdots \cup w_n) - w_i)^*$.

DEFINITION 2.5. A code event $R = (w_1 \cup \cdots \cup w_n)^*$ is unambiguous iff for all $w \in R$ if $w = w_{i_1} w_{i_2} \cdots w_{i_s} = w_{j_1} w_{j_2} \cdots w_{j_t}$, then $s = t$ and for each u $(1 \leqslant u \leqslant s)$, $i_u = j_u$; viz., every word in R can be parsed in only one way.

EXAMPLE 2.1. $(01 \cup 10)^*$ is unambiguous and $(00 \cup 000)^*$ is ambiguous.

3. A CLASSIFICATION OF CODE EVENTS

McNaughton and Papert introduced the notion of local parsability for unambiguous code events [1]. We extend the notion to ambiguous code events, by simply removing the restriction of unambiguity.

DEFINITION 3.1. Let k be a positive integer. For $w \in \Sigma^*$, let $P_k(w)$ $(S_k(w))$ be $L_k(w)$ $(R_k(w))$ in case $|w| \geqslant k$, or w, in case $|w| < k$. A code event R is k-parsable iff for all $w \in R$ and $x, y \in \Sigma^*$ such that $w = xy$, one can tell whether $x, y \in R$ or not, by knowing only $S_k(x)$ and $P_k(y)$. A code event is locally parsable iff it is k-parsable for some $k \geqslant 1$.

LEMMA 3.1. *If a code event R is not k-parsable, then there exist u, v, u_1, $u_2 \in \Sigma^*$ such that $|v| = k$ and either vu_1, $uvu_2 \in R$ but $vu_2 \notin R$, or else u_1v, $u_2vu \in R$ but $u_2v \notin R$.*

Proof. Assume R is not k-parable. Then there exist $w \in R$ and $x, y \in \Sigma^*$ such that $w = xy$ and one cannot tell whether $x, y \in R$ or not by knowing $S_k(x)$ and $P_k(y)$. Assume one cannot tell whether $y \in R$ or not. (In case one cannot tell whether $x \in R$ or not, the argument is symmetric). Then $|y| \geqslant k$. Furthermore, there exist $w' \in R$ and x', $y' \in \Sigma^*$ such that $w' = x'y'$, $P_k(y') = P_k(y)$ and either $y \in R$ but $y' \notin R$, or else $y \notin R$ but $y' \in R$. We may assume without loss of generality that $y \in R$ but $y' \notin R$. Let $P_k(y) = v$, $y = vu_1$, $y' = vu_2$ and $x' = u$. Then the result follows.

EXAMPLE 3.1. $(10101 \cup 1001)^*$ is 1-parable and $(00 \cup 000)^*$ is 2-parsable.

Notation. Let R be a code event. For any $w \in R$, let $w = {}_Rx - y$ or $w = x - y$, when the reference to R is understood, mean that $w = xy$ and $x, y \in R$. $w = x - y - z$ means $w = xyz$ and x, y, $z \in R$.

McNaughton and Papert gave the following three theorems [1].

THEOREM [1]-1. *For every $w \in \Sigma^+$, if $w = v^k$ for $k \geqslant 2$, then $w^* \notin NC$; if $w \neq v^k$, for any v and $k \geqslant 2$, then $w^* \in LT$.*

THEOREM [1]-2. *If an unambiguous code event $(w_1 \cup \cdots \cup w_n)^*$ is k-parsable, then it is $(2p + 2k - 1)$-testable, where $p = \max\{|w_i|\}$.*

THEOREM [1]-3. *If a locally testable code event* $(w_1 \cup \cdots \cup w_n)^*$ *satisfies one of the following, then it is locally parsable.*

(1) *For all* $i \neq j$, $1 \leqslant i, j \leqslant n$, w_i *is not a prefix of* w_j.

(2) *For all* $i \neq j$, $1 \leqslant i, j \leqslant n$, w_i *is not a suffix of* w_j.

In the rest of this section, we obtain generalizations over these theorems, which give an answer to Q1, Q2, Q3 in Section 1.

First some properties of code events are presented.

The following proposition is in part credited to Brzozowski [5].

PROPOSITION 3.1. *An event* R *is a code event iff* $R = R^*$ *and the event* $R - (R - \lambda)^2$ *is finite.*

Proof. If $R = (w_1 \cup \cdots \cup w_n)^*$, then $R = R^*$ and $R - (R - \lambda)^2 = (\lambda \cup w_1 \cup \cdots \cup w_n)$. Conversely, assume $R = R^*$ and $R - (R - \lambda)^2 = (\lambda \cup w_1 \cup \cdots \cup w_n)$. Let $R' = (w_1 \cup \cdots \cup w_n)^*$. Then

$$R = R^* \supseteq (w_1 \cup \cdots \cup w_n)^* = R'.$$

Let $w \in R$. If $w = \lambda$, then $w \in R'$. If $w \neq \lambda$, let $H(w) = \max\{l_i \mid w = w_{i_1} w_{i_2} \cdots w_{i_{l_i}}, w_{i_j} \in R - \lambda, 1 \leqslant j \leqslant l_i\}$. One can easily prove that $w \in R'$ by induction on $H(w)$.

Proposition 3.1 provides an algorithm for determining whether an arbitrary regular event is a code event or not.

The following proposition is also given in [6].

PROPOSITION 3.2. *An event* $(w_1 \cup \cdots \cup w_n)^*$ *is unambiguous iff for all* $x, y, z \in \Sigma^*$, $x, xy, yz, z \in R$ *implies* $y \in R$.

Proof. Assume there exist $x, y, z \in \Sigma^*$ such that $x, xy, yz, z \in R$ and $y \notin R$. Then the word xyz can be parsed in two incompatible ways: $xy - z$ and $x - yz$. Conversely, assume R is ambiguous. Then there exists $w = xyz$ such that w can be parsed in two ways: $w = xy - z$ and $w = x - yz$, where $xy \neq x - y$. Then $x, xy, yz, z \in R$ and $y \notin R$.

Proposition 3.2 provides the following algorithm for testing ambiguity of a code event. Let $R = (w_1 \cup \cdots \cup w_n)^*$. Then R is unambiguous iff $R \backslash R \cap R/R \cap \bar{R} = \varnothing$, where $R \backslash R = \{w \mid vw \in R \text{ for some } v \in R\}$ and $R/R = \{w \mid wv \in R \text{ for some } v \in R\}$. Note that if R is accepted by an automaton $A = \langle \Sigma, Q, \delta, S, F \rangle$, then $R \backslash R$ is accepted by the automaton $A' = \langle \Sigma, Q, \delta, F, F \rangle$. Thus $R \backslash R$ is regular. By symmetry, R/R is regular also. Thus from R one can effectively obtain a finite automaton for $R \backslash R \cap R/R \cap \bar{R}$, and test for emptiness.

An algorithm for testing ambiguity of a code event is also given in [7].

Now we shall obtain conditions for a code event to be noncounting.

DEFINITION 3.2. A regular event R has a general cyclic word iff there exist $x \in \Sigma^*$, $l \geqslant 0$, and $m \geqslant 2$ such that for all $i \geqslant l$, $x^i \in R$ iff $i \equiv 0 \pmod{m}$. Furthermore if $l = 0$, then R has a cyclic word. x is called a (general) cyclic word of R.

Remark. Let $\#Q$ be the number of states in the reduced automaton accepting R. Then in Definition 3.2, l can be replaced by $\#Q$.

DEFINITION 3.3. A regular event R has a pseudocyclic word iff there exist $x \in \Sigma^*$ and $m \geqslant 2$ such that $x \notin R$ and $x^m \in R$. x is called a pseudocyclic word of R.

We need the following version of the theorem due to Schützenberger [3] and McNaughton and Papert [1] on syntactic semigroups of noncounting events.

THEOREM 3.1. *A regular event R is not noncounting iff there exist $x, y, z \in \Sigma^*$ and $m \geqslant 2$ such that $x(y^m)^*z \subseteq R$ and $x(y^m)^*yz \cap R = \varnothing$.*

Proof. Necessity. Assume $R \notin \mathrm{NC}$. Let $A = \langle \Sigma, Q, \delta, S_1, F\rangle$ be the reduced automaton accepting R. There exist $y \in \Sigma^*$ and $m \geqslant 2$ states, $q_0, q_1, \ldots, q_{m-1}$ such that for $0 \leqslant i < m$, $\delta(q_0, y^i) = q_i$, and $\delta(q_0, y^m) = q_0$ (see [1]). Let R_i be the event accepted by the automaton $A_i = \langle \Sigma, Q, \delta, q_i, F\rangle$. Considering the sequence $(R_0, R_1, \ldots, R_{m-1}, R_0)$, one can see that there exist $q_{i_0}, q_{i_1} \in Q$ $(0 \leqslant i_0, i_1 < m)$ such that $\delta(q_{i_0}, y) = q_{i_1}$ and $R_{i_0} - R_{i_1} \neq \varnothing$. Let $z \in R_{i_0} - R_{i_1}$. Let $\delta(S_1, x) = q_{i_0}$. Then for all $i \geqslant 0$, $\delta(S_1, xy^{mi}z) \in F$ and $\delta(S_1, xy^{mi+1}z) \notin F$. Hence $x(y^m)^*z \subseteq R$ and $x(y^m)^*yz \cap R = \varnothing$.

Sufficiency. Assume $x(y^m)^*z \subseteq R$ and $x(y^m)^*yz \cap R = \varnothing$. For all $k \geqslant 0$, $x(y^m)^{k+1}z = xy^{mk+m-k} \cdot y^k \cdot z \in R$ and $xy^{mk+m-k} \cdot y^{k+1} \cdot z \notin R$. Hence $R \notin \mathrm{NC}$.

Now code event versions of Theorem 3.1 are given.

THEOREM 3.2. *A code event R is noncounting iff R has no general cyclic word.*

Proof. Necessity is obvious from Theorem 3.1.

Sufficiency. Assume $R \notin NC$. From Theorem 3.1, there exist $x, y, z \in \Sigma^*$ and $m \geqslant 2$ such that $x(y^m)^*z \subseteq R$ and $x(y^m)^*yz \cap R = \varnothing$. For sufficiently large k, consider the word $w = xy^{km}z$ in R. There exists a decomposition of w, $w = xy^{m_0}y_0 - y_1y^{m_1}y_0 - y_1y^{m_2}z$ such that $y = y_0y_1$. Let $v = y_1y_0$ and $p = m(m_1+1)$. Then $v^{m_1+1} = y_1y^{m_1}y_0 \in R$ and $p \geqslant 2$. Thus $v^{pi} \in R$ for all $i \geqslant 0$. Assume $v^{pi+1} \in R$ for some $i \geqslant 0$. Then $v^{(pi+1)(m_1+2)} \in R$. Thus

$$xy^{m_0}y_0(y_1y_0)^{(pi+1)(m_1+2)}y_1y^{m_2}z \in R.$$

But now $m_0 + (pi+1)(m_1+2) + m_2 + 1 \equiv 1 \pmod{m}$, which is a contradiction. Hence $(v^p)^*v \cap R = \varnothing$. Let $A = \langle \Sigma, Q, \delta, S_1, F\rangle$ be the reduced automaton accepting R. There exist $t \geqslant 0$ and $l \geqslant 1$ such that $\delta(S_1, v^{pt}) = \delta(S_1, v^{p(t+l)})$. Let $R_t = \{w \mid v^{pt}w \in R\}$ and $R_{t+1} = \{w \mid v^{p(t+1)}w \in R\}$. Note that $R_t = \{w \mid v^{p(t+l)}w \in R\}$.

If $w \in R_t$, then $v^{p(t+1)}w = v^p \cdot v^{pt}w \in R$. Conversely, if $w \in R_{t+1}$, then $v^{p(t+l)}w = v^{p(l-1)}v^{p(t+1)}w \in R$. Thus $R_t = R_{t+1}$. Hence $\delta(S_1, v^{p(t+1)}) = \delta(S_1, v^{pt})$ $(\in F)$. Let $q_j = \delta(S_1, v^{pt+j})$ for $0 \leqslant j \leqslant p$. Let $r = \min\{j \geqslant 1 \mid q_j \in F \text{ or } j = p\}$. Since $\delta(S_1, v^{pt+1}) \notin F$, $r \geqslant 2$. We claim that for all $i \geqslant pt$, $v^i \in R$ iff $i \equiv 0 \pmod r$. To see this, let $p = p_0 r + r_p$ $(0 \leqslant r_p < r)$. Since $v^{(pt+r)(p_0+1)} \in R$ and $(pt + r)(p_0 + 1) \equiv p_0 r + r \equiv r - r_p \pmod p$, $\delta(S_1, v^{(pt+r)(p_0+1)}) = q_{r-r_p} \in F$. Thus $r_p = 0$. Let $q_j \in F$ and $j = j_0 r + r_j$ $(r < j < p, 0 \leqslant r_j < r)$. Since $v^{pt+j+(pt+r)(p_0-j_0)} \in R$ and $pt + j + (pt + r)(p_0 - j_0) \equiv r_j \pmod p$, $r_j = 0$ by a similar argument. Conversely, let $r < j < p$ and $j = j_0 r$ for $j_0 \geqslant 0$. Then $q_j = \delta(S_1, v^{(pt+r)j_0}) \in F$. This completes the proof.

Example 3.2. $(0^4 \cup 0^6)^*$ is not noncounting, since $v = 0$, with $l = 4$ and $m = 2$, satisfies the condition for a general cyclic word.

Example 3.3. The following events have no general cyclic word, although they are not noncounting:

$$(00)^* \cup (000)^*, \qquad (1 \cup 00)^*(000)^*, \qquad 1(00)^*.$$

The following two corollaries present conditions for an unambiguous code event to be noncounting.

Corollary 3.1. *An unambiguous code event R is noncounting iff R has no cyclic word.*

Proof. Necessity is obvious from Theorem 3.1.

Sufficiency. Assume $R \notin NC$. From Theorem 3.2, there exist $x \in \Sigma^*$, $l \geqslant 0$, and $m \geqslant 2$ such that for all $i \geqslant l$, $x^i \in R$ iff $i \equiv 0 \pmod m$. Let $x^i \in R$ for some $i \geqslant 0$. Since $x^{ml+i} \in R$, $ml + i \equiv 0 \pmod m$. Thus $i \equiv 0 \pmod m$. Conversely, let $i = km$ for some $k \geqslant 0$. If $k = 0$, then $x^{km} = \lambda \in R$. If $k \geqslant 1$, then $x^{(l+k)m} = x^{lm} \cdot x^{km} = x^{km} \cdot x^{lm} \in R$ and $x^{lm} \in R$. From Proposition 3.2, $x^i = x^{km} \in R$.

Corollary 3.2. *An unambiguous code event R is noncounting iff R has no pseudocyclic word.*

Proof. Sufficiency is obvious from Theorem 3.2.

Necessity. Assume $R \in \mathrm{NC}$ and R has a pseudocyclic word. Then there exist $x \in \Sigma^*$, $m \geqslant 2$, and $k \geqslant 0$ such that $x \notin R$, $x^m \in R$ and for all $v_1, v_2, v_3 \in \Sigma^*$, $v_1 v_2^k v_3 \in R$ iff $v_1 v_2^{k+1} v_3 \in R$. Since $x^m \in R$, $x^{km} = x^{k(m-1)} \cdot x^k \in R$. Thus $x^{km+1} = x^{k(m-1)} \cdot x^{k+1} \in R$. But $x^{km+1} = x \cdot x^{km} = x^{km} \cdot x$. From Proposition 3.2, $x \in R$, which is a contradiction.

Example 3.4. Let $R_1 = (010 \cup 101)^*$, $R_2 = (01 \cup 10 \cup 11)^*$, and $R_3 = (01 \cup 10)^*$. These are all unambiguous. Moreover, $01 \notin R_1$, $(01)^3 \in R_1$, $01011 \notin R_2$, $(01011)^2 \in R_2$, and R_3 has no pseudocyclic word. Thus $R_1, R_2 \notin \mathrm{NC}$ and $R_3 \in \mathrm{NC}$.

Remark. There exist ambiguous code events which are noncounting but have a pseudocyclic word: $(00 \cup 000)^*$.

Note that in Examples 3.2 and 3.4, $m \leqslant \max\{| w_i |\}$. We assert without proof that this inequality holds for any code event R and any shortest (general, or pseudo-) cyclic word of R.

Theorem 3.2, Corollary 3.1, or Corollary 3.2 does not seem to provide a new algorithm for determining whether an arbitrary code event is noncounting or not, although they may be regarded as a generalization of the first part of Theorem [1]-1.

Next we obtain conditions for a code event to be locally parsable, which turn out to be equivalent to those for strictly local testability.

THEOREM 3.3. *An event $R = (w_1 \cup \cdots \cup w_n)^*$ is not locally parsable iff there exist $x, y \in \Sigma^*$ such that $xy, yx \in R$ and $x(yx)^* \cap R = \varnothing$.*

Proof. Sufficiency. Assume R is k-parsable and there exist $x, y \in \Sigma^*$ such that $xy, yx \in R$ and $x(yx)^* \cap R = \varnothing$. Consider $w = v_1 v_2$ and $w' = v_1' v_2'$, where $v_1 = x(yx)^k$, $v_2 = (yx)^k y$, and $v_1' = v_2' = (yx)^k$. Then $w, w' \in R$, $R_k(v_1) = R_k(v_1')$, $L_k(v_2) = L_k(v_2')$, and $v_1', v_2' \in R$. But $v_1 \notin R$, which is a contradiction.

Necessity. Assume R is not locally parsable. Let $k = 2p^2n^2r + p$, where $p = \max\{| w_i |\}$, $r = \# Q$ and $A = \langle \Sigma, Q, \delta, S_1, F\rangle$ is the reduced automaton accepting R. From Lemma 3.1, there exist $v_1, v_2 \in R$ and $v \in \Sigma^*$ such that either $v = L_k(v_1) = L_k(v_2)$ or $v = R_k(v_1) = R_k(v_2)$ and v can be decomposed in two ways according to the parsings of v_1 and v_2. Assume $v = L_k(v_1) = L_k(v_2)$. (In case $v = R_k(v_1) = R_k(v_2)$, the result follows in a similar way.) Let two decompositions of v according to the parsings of v_1 and v_2 be as follows: $v = w_{i_1} w_{i_2} \cdots w_{i_s} z_1$ (Decomposition A), and $v = y_0 w_{j_1} w_{j_2} \cdots w_{j_t} z_2$ (Decomposition B), where $y_0 \in \{w \mid w_i = xw$ for some i, $1 \leqslant i \leqslant n$, and some $x \in \Sigma^+\}$, $z_1, z_2 \in \{w_i = wx$ for some i, $1 \leqslant i \leqslant n$, and some $x \in \Sigma^+\}$ and for each $0 \leqslant \mu \leqslant t$, $y_0 w_{j_1} \cdots w_{j_\mu} \notin R$.

Construct the sequence, W, of triples $(w_{i_\nu}, l, w_{j_\mu})$, where $1 \leqslant \nu \leqslant s$, $1 \leqslant \mu \leqslant t$ and $0 < l < p$, according to Decompositions A and B as follows: (1) W begins with the triple (w_{i_ν}, l, w_{j_1}) such that $\nu = \min\{m \mid | w_{i_1} w_{i_2} \cdots w_{i_m} | > | y_0 |\}$ and

$$l = | y_0 | - | w_{i_1} w_{i_2} \cdots w_{i_{\nu-1}} |$$

or $l = | y_0 |$ if $\nu = 1$. (2) The triple $(w_{i_{\nu'}}, l', w_{j_{\mu'}})$ immediately after the triple $(w_{i_\nu}, l, w_{j_\mu})$ is the one such that

$$\mu' = \min\{m + 1 \mid | w_{i_1} w_{i_2} \cdots w_{i_\nu} | < | y_0 w_{j_1} \cdots w_{j_m} |\},$$

$$\nu' = \min\{m \mid | w_{i_1} w_{i_2} \cdots w_{i_m} | > | y_0 w_{j_1} \cdots w_{j_{\mu'-1}} |\},$$

and

$$l' = |\, y_0 w_{j_1} \cdots w_{j_{\mu'-1}} \,| - |\, w_{i_1} w_{i_2} \cdots w_{i_{\nu'-1}} \,|.$$

We note that the two parsings of v do not have a common parsing mark. (For otherwise, $y_0 w_{j_1} \cdots w_{j_\mu} \in R$ for some $\mu \leqslant t$.) For each triple $(w_{i_\nu}, l, w_{j_\mu})$ in W, the left boundary of w_{j_μ} is between the left and right boundaries of w_{i_ν}, and l is a positive integer. Furthermore, if $(w_{i_{\mu''}}, l'', w_{j_{\mu''}})$ follows $(w_{i_\nu}, l, w_{j_\mu})$ in W, then the left boundaries of both $w_{i_{\nu''}}$ and $w_{j_{\mu''}}$ are to the right of the right boundaries of both w_{i_ν} and w_{j_μ}.

Construct W as long as possible. It is not difficult to see that at least one triple $(w_{i_\nu}, l, w_{j_\mu})$ appears more than r times in W. Thus v can be decomposed in the following two ways:

$$v = v_0 w_{i_\nu} v_1 w_{i_\nu} v_2 \cdots w_{i_\nu} v_r w_{i_\nu} v_{r+1}$$

and

$$v = v_0' w_{j_\mu} v_1' w_{j_\mu} v_2' \cdots w_{j_\mu} v_r' w_{j_\mu} v_{r+1}',$$

where $v_0 \in R$, v_0', $v_0' w_{j_\mu} \notin R$ and for each $1 \leqslant m \leqslant r$, $v_m, v_m' \in R$, $v_0' w_{j_\mu} v_1' \cdots w_{j_\mu} v_m'$, $v_0' w_{j_\mu} v_1' \cdots v_m' w_{j_\mu} \notin R$, and $l = |\, v_0' w_{j_\mu} v_1' \cdots w_{j_\mu} v_m' \,| - |\, v_0 w_{i_\nu} v_1 \cdots w_{i_\nu} v_m \,| = |\, v_0' \,| - |\, v_0 \,|$. Let $q_m = \delta(S_1, v_0 w_{i_\nu} v_1 \cdots w_{i_\nu} v_m)$ for $0 \leqslant m \leqslant r$. Then $q_m \in F$. Since $r = \#Q$, there exist m_0 and m_1 such that $0 \leqslant m_0 < m_1 \leqslant r$ and $q_{m_0} = q_{m_1}$. Let $w = v_0 w_{i_\nu} v_1 w_{i_\nu} \cdots v_{m_0}$, $v_0' = v_0 x$, $w_{i_\nu} = xx'$ and $y = x' v_{m_0+1} w_{i_\nu} \cdots w_{i_\nu} v_{m_1}$. Then $w \in R$, $xy = w_{i_\nu} v_{m_0+1} \cdots w_{i_\nu} v_{m_1} \in R$, $yx = w_{j_\mu} v'_{m_0+1} \cdots w_{j_\mu} v'_{m_1} \in R$ and $\delta(S_1, w(xy)^* x) = \{\delta(S_1, wx)\}$. Since $wx \notin R$ and $w \in R$, $x(yx)^* \cap R = \varnothing$. This completes the proof.

Corollary 3.3. *If an event $R = (w_1 \cup \cdots \cup w_n)^*$ is locally parsable, then R is k-parsable, where $k = 2p^2n^2r + p$, $p = \max\{|\, w_i \,|\}$, and r is the number of states in the reduced automaton accepting R.*

Corollary 3.4. *An event $R = (w_1 \cup \cdots \cup w_n)^*$ is locally parsable iff the event $R\alpha \cap \alpha R \cap \bar{R}$ is finite, where $\alpha = \{w \mid w_i = wv$ for some i, $1 \leqslant i \leqslant n$, and $v \in \Sigma^+\}$.*

Proof. Assume R is not locally parsable. From the proof of Theorem 3.3, one can see that there exist $x, y \in \Sigma^+$ such that $xy, yx \in R$, $x(yx)^* \cap R = \varnothing$ and $x \in \alpha$. Then $x(yx)^* \subseteq R\alpha \cap \alpha R \cap \bar{R}$. Thus $R\alpha \cap \alpha R \cap \bar{R}$ is infinite. Conversely, assume $R\alpha \cap \alpha R \cap \bar{R}$ is infinite. Then there exist $v, z_1, y_0 \in \Sigma^*$ such that $|\, v \,| \geqslant 2p^2n^2r + p$, $v \notin R$, $z_1, y_0 \in \alpha$, and $v = wz_1 = y_0 w'$ for some $w, w' \in R$, where p, n, and r are as in the proof of Theorem 3.3. Since $v \notin R$, v can be decomposed in two ways according to the parsings of $w \cdot z_1$ and $y_0 \cdot w'$. As in the proof of Theorem 3.3, one can see that there exist $x, y \in \Sigma^*$ such that $xy, yx \in R$ and $x(yx)^* \cap R = \varnothing$. From Theorem 3.3, R is not locally parsable.

THEOREM 3.4. *A code event R is strictly locally testable iff R is locally parsable.*

Proof. It will suffice to prove the following two lemmas.

LEMMA 3.2. *If an event $R = (w_1 \cup \cdots \cup w_n)^*$ is k-parsable, then R is strictly $(2k + p)$-testable, where $p = \max\{| w_i |\}$.*

Proof. Let $m = 2k + p$. We shall prove that for all w of length $\geqslant m$, $w \in R$ iff $T_m(w) \in V_m(R)$. Let w be any word of length $\geqslant m$. If $w \in R$, then $T_m(w) \in V_m(R)$ by definition. Conversely assume $T_m(w) \in V_m(R)$. Let $| w | \geqslant m + 2$. (In case $| w | = m$ or $m + 1$, the proof is similar.) Decompose w as follows: $w = xa_1a_2z = b_1x'a_2z = w'a_3$, where $| x | = | x' | = m$ and $a_1, a_2, a_3, b_1 \in \Sigma$. We shall try to parse w with $\{w_1, ..., w_n\}$ from the left. $x = L_m(v)$ for some $v \in R$. Let $v = xv_1$. There exists a decomposition of x, $x = x_0x_1$, according to the parsing of v such that $x_0, x_1v_1 \in R$, $k < | x_0 | \leqslant k + p$ and $k \leqslant | x_1 | < k + p$. Let $x_0 = b_1z_0$. $x' \in I_m(v')$ for some $v' \in R$. Let $v' = v_0'x'v_1'$. Consider a decomposition of v', $v' = v_0'z_0x_1a_1v_1'$. Note that $z_0x_1a_1 = x'$. Since $v', x_0, x_1v_1 \in R$, $R_k(v_0'z_0) = R_k(x_0)$, $L_k(x_1a_1v_1') = L_k(x_1v_1)$, and R is k-parsable, it follows that $x_1a_1v_1' \in R$. Thus $x_0x_1a_1v_1' = b_1x'v_1' \in R$. Thus there exists a decomposition of b_1x', $b_1x' = x_0'x_1'$ such that $x_0', x_1'v_1' \in R$, $k + 1 < | x_0' | \leqslant k + p + 1$, and $k \leqslant | x_1' | < k + p$. Proceeding in this fashion, one eventually ends up with a decomposition of w', $w' = w_0'w_1'$ such that $w_0', w_1'z' \in R$ for some $z \in \Sigma^*$ and $k \leqslant | w_1' | < k + p$. Let $y = R_m(w)$. $y = R_m(v'')$ for some $v'' \in R$. Let $v'' = v_0''y = v_0''z_0''w_1'a_3$. Since $v'', w_0', w_1'z' \in R$, $R_k(v_0''z_0'') = R_k(w_0')$, $L_k(w_1'a_3) = L_k(w_1'z')$, and R is k-parsable, it follows that $w_1'a_3 \in R$. Hence $w = w_0'w_1'a_3 \in R$.

LEMMA 3.3. *If en event $R = (w_1 \cup \cdots \cup w_n)^*$ is strictly testable, then R is locally parsable.*

Proof. Assume R is strictly m-testable and not locally parsable. From Theorem 3.3, there exist $x, y \in \Sigma^*$ such that $xy, yx \in R$ and $x(yx)^* \cap R = \varnothing$. Consider $x(yx)^m$. Since $(xy)^m, (yx)^m, (xy)^{m+1} \in R$, $L_m(x(yx)^m) = L_m((xy)^m)$, $I_m(x(yx)^m) \subseteq I_m((xy)^{m+1})$, and $R_m(x(yx)^m) = R_m((yx)^m)$, it follows that $x(yx)^m \in R$, which is a contradiction. This completes the proof of Theorem 3.4.

Note that Lemma 3.2 provides a stronger version of Theorem [1]-2.

We next present a theorem on unambiguous, locally testable code events, which gives an affirmative answer to Question 3.

THEOREM 3.5. *Let R be an unambiguous code event. The following are equivalent.*

(1) *R is locally testable.*

(2) *R is strictly locally testable.*

(3) *R is locally parsable.*[1]

[1] The equivalence of (2) and (3) is also proved by Restivo [11].

Proof. The equivalence of (2) and (3) is obvious from Theorem 3.4. (2) $\Rightarrow$ (1) is obvious by definition. Thus it will suffice to prove (1) $\Rightarrow$ (3). Assume R is k-testable and not locally parsable. From Theorem 3.3, there exist $x, y \in \Sigma^*$ such that $xy, yx \in R$ and $x(yx)^* \cap R = \varnothing$. Let $v_1 = (xy)^k(yx)^k(xy)^k$, $v_2 = (xy)^k(yx)^k(xy)^k x(xy)^k$, and $v_3 = (xy)^k x(xy)^k(yx)^k(xy)^k$. Since $T_k(v_1) = T_k(v_2) = T_k(v_3)$, $v_1 \in R$, and R is k-testable, $v_2, v_3 \in R$. Moreover, since $(xy)^k(yx)^k$, $(yx)^k(xy)^k \in R$, it follows that $(xy)^k x(xy)^k = x(yx)^k(xy)^k \in R$ from Proposition 3.2. Now let $v_4 = (yx)^k(xy)^k(yx)^k$, $v_5 = (yx)^k(xy)^k(yx)^k x(yx)^k$, and $v_6 = (yx)^k x(yx)^k(xy)^k(yx)^k$. Considering v_4, v_5, and v_6, one can see that $(yx)^k x(yx)^k = (yx)^k(xy)^k x \in R$, by the same argument as above. Moreover, since $(yx)^k(xy)^k \in R$, it follows that $x \in R$, from Proposition 3.2. This is a contradiction.

Remark. There exist ambiguous code events which are locally testable, but not strictly locally testable: Let

$$R = (10 \cup 01 \cup 00 \cup 11 \cup 000 \cup 111 \cup 001 \cup 110 \cup 011 \cup 100)^*.$$

Then $\bar{R} = 0(10)^* \cup 1(01)^*$ and $T_2(\bar{R}) = \{\langle 01, \phi, 10\rangle, \langle 10, \phi, 01\rangle, \langle 01, 10 \cup 01, 10\rangle, \langle 10, 01 \cup 10, 01\rangle\}$. One can see that for all $w \in \Sigma^*$ of length $\geqslant 2$, $w \in \bar{R}$ iff $T_2(w) \in T_2(\bar{R})$. Thus $\bar{R}$ is 2-testable and so is R. Let $x = 1$ and $y = 0$. Then $xy, yx \in R$ and $x(yx)^* \cap R = \varnothing$. From Theorems 3.3 and 3.4, R is neither locally parsable, nor strictly locally testable.

We utilize the following theorem which was proved by McNaughton, Zalcstein, Brzozowski, and Simon, to obtain conditions for ambiguous code events to be locally testable.

THEOREM. *$R \subseteq \Sigma^+$ is locally testable iff for all idempotent $e \in S$, eSe is a commutative semigroup in which every element is idempotent, where S is the semigroup of transformations on the states of the reduced automaton accepting R induced by the words of Σ^+ and an element $e \in S$ is idempotent iff $e^2 = e$.*

Remark. It is well known that S is isomorphic to the quotient of Σ^+ modulo the equivalence relation $\equiv$ defined by $w_1 \equiv w_2$ iff for all $v, w \in \Sigma^*$, $vw_1w \in R$ iff $vw_2w \in R$ (see [1, 2]). For $w \in \Sigma^+$, let $[w]$ denote the equivalence class of w modulo $\equiv$. Now let e be a word such that $[e]^2 = [e]$ in S. One can easily see that for all $x, y \in \Sigma^*$, $xee^*y \subseteq R$ iff $xey \in R$, and $xee^*y \cap R = \varnothing$ iff $xey \notin R$.

THEOREM 3.6. *A code event R is not locally testable iff there exist $x, y, z \in \Sigma^*$ and $v, w \in R$ such that one of the following holds.*

(1) $xy, yx \in R$ *and* $(xy)^*vy(xy)^*v(yx)^* \cap R = \varnothing$.

In the following, assume xyz, yzx, $zxy \in R$.

(2) $(xyz)^*v(yzx)^*yzvy(zxy)^* \subseteq R$ *and* $(xyz)^*vy(zxy)^* \cap R = \varnothing$.

(3) $(xyz)^*v(zxy)^*zvzx(yzx)^* \subseteq R$ *and* $(xyz)^*vzx(yzx)^* \cap R = \varnothing$.

(4) $(xyz)^*xv(zxy)^*zw(yzx)^*y \cap R = \varnothing$.

(5) $(xyz)^*xyv(yzx)^*yzw(zxy)^*zx \cap R = \varnothing$.

Proof. Sufficiency. It is easy to see that in each case where one of the five conditions holds, there exist $w, w' \in \Sigma^*$ such that $T_k(w) = T_k(w')$, $w \in R$, and $w' \notin R$ for each $k \geqslant 1$. (For example, in case (1), let $w = (xy)^k v(yx)^k$ and $w' = (xy)^k vy(xy)^k v(yx)^k$.)

Necessity. Assume R is not locally testable. From the McNaughton–Zalcstein theorem and the remark above, one can see that there exist $v_1, e, v', w', v_2 \in \Sigma^*$ such that one of the following holds.

(a) $v_1ee^*v'ee^*v_2 \subseteq R$ and $v_1ee^*v'ee^*v'ee^*v_2 \cap R = \varnothing$.

(b) $v_1ee^*v'ee^*v_2 \cap R = \varnothing$ and $v_1ee^*v'ee^*v'ee^*v_2 \subseteq R$.

(c) $v_1ee^*v'ee^*w'ee^*v_2 \subseteq R$ and $v_1ee^*w'ee^*v'ee^*v_2 \cap R = \varnothing$.

We shall prove (c) implies (4) or (5). (It can be verified in a similar way that (a) implies (1), and (b) implies (2) or (3).)

Assume (c). For sufficiently large k, consider $w_0 = v_1e^kv'e^kw'e^kv_2$ $(\in R)$. It is not difficult to see that there exists a decomposition of

$$w_0, \; w_0 = v_1e^{k_0}e_{11} - (e_{12}e_{11})^{k_1} - e_{12}e^{k_2}v'e^{k_3}e_{21} - (e_{22}e_{21})^{k_4} - e_{22}e^{k_5}w'e^{k_6}e_{31} - (e_{32}e_{31})^{k_7} - e_{32}e^{k_8}v_2$$

such that $e = e_{11}e_{12} = e_{21}e_{22} = e_{31}e_{32}$. Let $m = k_1k_4k_7$,

$$w = (e_{12}e_{11})^m \, e_{12}e^{k_2}v'e^{k_3}e_{21}, \qquad \text{and} \qquad v = (e_{22}e_{21})^m \, e_{22}e^{k_5}w'e^{k_6}e_{31}.$$

Then $(e_{12}e_{11})^m$, $(e_{22}e_{21})^m$, $(e_{32}e_{31})^m$, $v, w \in R$. Since $e_{11}e_{12} = e_{21}e_{22} = e_{31}e_{32}$, there exist $x, y, z \in \Sigma^*$ such that one of the following holds.

(c1) $(e_{12}e_{11})^m = xyz$, $(e_{22}e_{21})^m = yzx$, and $(e_{32}e_{31})^m = zxy$,

(c2) $(e_{12}e_{11})^m = xyz$, $(e_{22}e_{21})^m = zxy$, and $(e_{32}e_{31})^m = yzx$.

Assume (c1). It is not difficult to see

$$\begin{aligned}(xyz)^* \, & xv(zxy)^* \, zw(yzx)^* \, y \\ &= (xyz)^* \, x(e_{22}e_{21})^m \, e_{22}e^{k_5}w'e^{k_6}e_{31}(zxy)^* \, z(e_{12}e_{11})^m \, e_{12}e^{k_2}v'e^{k_3}e_{21}(yzx)^* \, y \\ &\subseteq e_{12}ee^*w'ee^*v'ee^*e_{31}.\end{aligned}$$

From (c), $e_{12}ee^*w'ee^*v'ee^*e_{31} \cap R = \varnothing$. Thus $(xyz)^*xv(zxy)^*zw(yzx)^*y \cap R = \varnothing$. In case (c2), it follows in a similar way that $(xyz)^*xyv(yzx)^*yzw(zxy)^*zx \cap R = \varnothing$.

Remark. Each condition in the above theorem implies the existence of x' and y' such that $x'y'$, $y'x' \in R$ and $x'(y'x')^* \cap R = \varnothing$.

4. Homomorphisms Which Preserve NC, LT, and SLT, Respectively

Throughout this section, let f be a homomorphism from the set of words over an alphabet Σ_1 to the set of words over another alphabet Σ_2. Extend f to the class of events over Σ_1 as follows. For any event $R \subseteq \Sigma_1^*$, $f(R) = \{w \mid w = f(v) \text{ for some } v \in R\}$.

Definition 4.1. A homomorphism f preserves NC, LT, and SLT iff f satisfies the following conditions (1), (2), and (3), respectively.

(1) For all $R \subseteq \Sigma_1^*$, $R \in \mathrm{NC}$ iff $f(R) \in \mathrm{NC}$.

(2) For all $R \subseteq \Sigma_1^*$, $R \in \mathrm{LT}$ iff $f(R) \in \mathrm{LT}$.

(3) For all $R \subseteq \Sigma_1^*$, $R \in \mathrm{SLT}$ iff $f(R) \in \mathrm{SLT}$.

Proposition 4.1. *For all $R \subseteq \Sigma_1^*$, an injective homomorphism f satisfies the following.*

(1) *If $R \notin \mathrm{NC}$, then $f(R) \notin \mathrm{NC}$.*

(2) *If $R \notin \mathrm{LT}$, then $f(R) \notin \mathrm{LT}$.*

(3) *If $R \notin \mathrm{SLT}$, then $f(R) \notin \mathrm{SLT}$.*

Proof. (1) and (2) are presented in [1], so only (3) will be proved. Assume $f(R)$ is strictly k-testable. Let $m = k + 2$. We shall prove for all $w \in \Sigma_1^*$ of length $\geqslant m$, $w \in R$ iff $T_m(w) \in V_m(R)$. Let $w \in \Sigma_1^*$ of length $\geqslant m$. If $w \in R$, then, by definition, $T_m(w) \in V_m(R)$. Conversely, assume $T_m(w) \in V_m(R)$. Then $L_m(w) = L_m(y_0)$ and $R_m(w) = R_m(y_1)$ for some $y_0, y_1 \in R$. Consider $f(w)$. Note that f being injective and $f(\lambda) = \lambda$ imply that f is length nondecreasing, i.e., $|f(w)| \geqslant |w|$, for all $w \in \Sigma_1^*$. Then $L_k(f(w)) = L_k(f(y_0)) \in L_k(f(R))$, $R_k(f(w)) = R_k(f(y_1)) \in R_k(f(R))$,

$$I_k(f(L_{k+1}(w))a) \cup I_k(af(R_{k+1}(w))) \subseteq I_k(f(y_0)) \cup I_k(f(y_1)) \subseteq I_k(f(R)),$$

where $a \in \Sigma_2$. Let $\beta_0 = \{x \mid x \in I_k(y) \text{ for some } y \in af(I_k(w))a\}$. It is easy to see that $\beta_0 \subseteq I_k(f(R))$. Thus $I_k(f(w)) = I_k(f(L_{k+1}(w))a) \cup \beta_0 \cup I_k(af(R_{k+1}(w))) \subseteq I_k(f(R))$. Thus $f(w) \in f(R)$. Hence $w \in R$.

Remark. From the proof of Proposition 4.1 and the isomorphism f such that $\Sigma_1 = \Sigma_2$ and for all $a \in \Sigma_1$, $f(a) = a$, one can see that if $R \subseteq \Sigma^*$ is strictly k-testable, then it is strictly $(k + i)$-testable for all $i \geqslant 2$. Note that the lower bound of i is critical. There exist strictly k-testable events which are not $(k + 1)$-testable: Let $R = 0^k \cup 0^{k+1}$. Obviously R is strictly k-testable. But R is not $(k + 1)$-testable since $T_{k+1}(0^{k+1}) = T_{k+1}(0^{k+2}) = \langle 0^{k+1}, \phi, 0^{k+1} \rangle$, $0^{k+1} \in R$, and $0^{k+2} \notin R$.

PROPOSITION 4.2. *An injective homomorphism f preserves* NC *iff* $f(\Sigma_1^*) \in$ NC.

Proof. Necessity. Obvious from $\Sigma_1^* \in$ NC.

Sufficiency. Assume $f(\Sigma_1^*) \in$ NC. From Proposition 4.1, it will suffice to prove that $R \in$ NC implies $f(R) \in$ NC for all $R \subseteq \Sigma_1^*$. It is well known that an event $R \subseteq \Sigma^*$ is noncounting iff R belongs to the smallest class of events containing a_i, $a_i \in \Sigma$, and λ and closed under concatenation and the Boolean operations (see [1]). Using this fact, the proposition is proved by induction on the number k of operators ($\cup$, $-$, $\cdot$) contained in R. If $R = a_i$ or λ, where $a_i \in \Sigma_1$, then obviously $f(R) \in$ NC. Assume the assertion holds for all noncounting R with at most k operators. Now let R have $k + 1$ operators. We have the following cases.

(1) $R = R_1 \cup R_2$, $\quad f(R) = f(R_1) \cup f(R_2)$,
(2) $R = R_1 R_2$, $\quad f(R) = f(R_1) f(R_2)$,
(3) $R = \bar{R}_1$, $\quad f(R) = f(\Sigma_1^*) - f(R_1)$.

In every case it is easy to see that $f(R) \in$ NC.

THEOREM 4.1. *An injective homomorphism f preserves* SLT *iff* $f(\Sigma_1^*) \in$ SLT.

Proof. Necessity. Obvious from $\Sigma_1^* \in$ SLT.

Sufficiency. Assume $f(\Sigma_1^*) \in$ SLT. From Theorem 3.4, $f(\Sigma_1^*)$ is k-parsable for some $k \geqslant 1$. From Proposition 4.1, it will suffice to prove $R \in$ SLT implies $f(R) \in$ SLT for all $R \subseteq \Sigma_1^*$. Assume R is strictly l-testable. Let $m = 2k + lp$, where $p = \max\{|w_i| \mid w_i \in f(\Sigma_1)\}$.

We shall prove that for all $w \in \Sigma_2^*$ of length $\geqslant m$, $w \in f(R)$ iff $T_m(w) \in V_m(f(R))$. Let w be any word in Σ_2^* of length $\geqslant m$. If $w \in f(R)$, then $T_m(w) \in V_m(f(R))$ by definition. Conversely, assume $T_m(w) \in V_m(f(R))$. One can verify that $w = f(v)$ for some $v \in \Sigma_1^*$ by the argument similar to the one for Lemma 3.2. If will suffice to show $v \in R$. Let $v = L_l(v)v_0 \,.\, L_m(w) = L_m(y)$ for some $y \in f(R)$. Let $y = f(L_l(v))y_0$. Since $f(L_l(v))$, $f(v_0) \in f(\Sigma_1^*)$, $L_k(y_0) = L_k(f(v_0))$, and $f(\Sigma_1^*)$ is k-parsable, it follows that $y_0 \in f(\Sigma_1^*)$. Thus $L_l(v) \in L_l(R)$. Similarly, $R_l(v) \in R_l(R)$. Now let $x \in I_l(v)$ and $v = v_0 x v_1$. We have three cases:

(1) $|f(v_0)|, |f(v_1)| > k$. There exist $y' \in f(R)$, $z_0, z_0', z_1, z_1' \in \Sigma_2^*$ such that $z_0 f(x) z_1 \in I_m(y')$, $w = z_0' z_0 f(x) z_1 z_1'$, and $|z_0|, |z_1| \geqslant k$. Let $y' = y_0' z_0 f(x) z_1 y_1'$. Since $R_k(y_0' z_0) = R_k(z_0' z_0)$, $L_k(f(x) z_1 y_1') = L_k(f(x) z_1 z_1')$, and $f(\Sigma_1^*)$ is k-parsable, it follows that $y_0' z_0 \in f(\Sigma_1^*)$. Similarly, $z_1 y_1' \in f(\Sigma_1^*)$.

Thus $x \in I_l(R)$.

(2) $|f(v_0)| \leqslant k$. Then $|f(v_0 x)| \leqslant k + lp$. Recall that $L_m(w) = L_m(y)$ for $y \in f(R)$. Let $y = f(v_0 x) y_2$. Then $L_k(y_2) = L_k(f(v_1))$. Thus $y_2 \in f(\Sigma_1^*)$ by the similar reasoning as for the above.

Thus $x \in I_l(R)$.

(3) $f(v_1) \leqslant k$. The argument is the same as in case (2). From these observations it follows that $v \in R$. This completes the proof.

PROPOSITION 4.3. *If an injective homomorphism f preserves* SLT, *then f preserves* LT.

Proof. Assume f preserves SLT. From Proposition 4.1, it will suffice to prove $R \in \mathrm{LT}$ implies $f(R) \in \mathrm{LT}$ for all $R \subseteq \Sigma_1{}^*$. Note that LT is the Boolean closure of SLT, $f(R_1 \cup R_2) = f(R_1) \cup f(R_2)$, and $f(\bar{R}) = f(\Sigma_1{}^*) - f(R)$. Moreover, by assumption, $f(\Sigma_1{}^*) \in \mathrm{SLT}$. The proof can easily be done by induction on the number of Boolean operators.

COROLLARY 4.1. *The following are equivalent for an injective homomorphism f.*

(1) *$f(\Sigma_1{}^*)$ is locally testable.*

(2) *$f(\Sigma_1{}^*)$ is strictly locally testable.*

(3) *$f(\Sigma_1{}^*)$ is locally parsable.*

(4) *f preserves* LT.

(5) *f preserves* SLT.

Proof. The equivalence of (1), (2), and (3) is the content of Theorem 3.5. (4) $\Rightarrow$ (1) is obvious, since $\Sigma_1{}^* \in \mathrm{LT}$. From Theorem 4.1, (2) $\Rightarrow$ (5). From Proposition 4.3, (5) $\Rightarrow$ (4). Hence (4) $\Rightarrow$ (1) $\Rightarrow$ (2) $\Rightarrow$ (5) $\Rightarrow$ (4).

Finally, we shall obtain conditions for a homomorphism f to preserve NC, LT, and SLT, respectively. Let us say that a homomorphism f is nontrivial if $f(a) \neq \lambda$ for some $a \in \Sigma_1$. In the rest of the paper, we consider only nontrivial homomorphisms, which may cause no loss of generality.

DEFINITION 4.2. A homomorphism f preserves weakly NC, LT, and SLT iff f satisfies, respectively, the following.

(1) For all $R \subseteq \Sigma_1{}^*$, $R \in \mathrm{NC}$ implies $f(R) \in \mathrm{NC}$.

(2) For all $R \subseteq \Sigma_1{}^*$ $R \in \mathrm{LT}$ implies $f(R) \in \mathrm{LT}$.

(3) For all $R \subseteq \Sigma_1{}^*$, $R \in \mathrm{SLT}$ implies $f(R) \in \mathrm{SLT}$.

LEMMA 4.1. *If a nontrivial homomorphism f preserves weakly* NC, LT, *or* SLT, *then it is injective.*

Proof. Assume f is not injective and f preserves weakly NC, LT, or SLT. Let $p = \min\{|w| \mid w \in \Sigma_1{}^* \text{ and } f(w) = f(w') \text{ for some } w' \in \Sigma_1{}^* - \{w\}\}$. We have two cases:

(1) $p = 0$. Then $f(a) = \lambda$ for some $a \in \Sigma_1$. Since f is nontrivial, there exists $b \in \Sigma_1$ such that $f(b) \neq \lambda$. Let $R = (ab^2a)^*$. Then R is 1-parsable. From Theorem 3.4, $R \in \mathrm{SLT}$. But $f(R) = ((f(b))^2)^* \notin \mathrm{NC}$ from Theorem 3.1, which is a contradiction.

(2) $p \geqslant 1$. Let $w, w' \in \Sigma_1^+$ be words such that $|w| = p$, $w \neq w'$ and $f(w) = f(w')$. Let $R = (ww')^*$. Then $f(R) = ((f(w))^2)^*$. From Theorem 3.1, $f(R) \notin \mathrm{NC}$. Thus it will suffice to show that $R \in \mathrm{SLT}$. Assume $R \notin \mathrm{SLT}$. From Theorems 3.5 and [1]-1, $R \notin \mathrm{NC}$ and there exist $v \in \Sigma_1^+$ and $k \geqslant 2$ such that $ww' = v^k$. Let $w = v^{k_1}v_0$ and $w' = v_1 v^{k_2}$, where $v = v_0 v_1$ and $k = k_1 + k_2 + 1$. Since for all $a \in \Sigma_1$, $f(a) \neq \lambda$, $w \neq w'$, and $f(w) = f(w')$, there exist no $x, y \in \Sigma_1^*$ such that either $w = xw'y$ or else $w' = xwy$. Thus $k_1, k_2 \geqslant 1$, for otherwise $ww' = v$. And $v_0 v_1 \neq v_1 v_0$, for otherwise $w' = v_1(v_0 v_1)^{k_2} = (v_1 v_0)^{k_2} v_1 = (v_0 v_1)^{k_2} v_1 = wy'$ for some $y' \in \Sigma_1^*$, since $|w'| \geqslant |w|$. Furthermore

$$f(w) = f((v_0v_1)^{k_1} v_0) = f(v_0v_1) f((v_0v_1)^{k_1-1} v_0) = f(w') = f(v_1(v_0v_1)^{k_2})$$

$$= f(v_1v_0) f((v_1v_0)^{k_2-1} v_1).$$

Since $|f(v_0v_1)| = |f(v_1v_0)|$, it follows that $f(v_0v_1) = f(v_1v_0)$, which is a contradiction to $|v_0v_1| < |w|$.

THEOREM 4.2. *The following are equivalent for a nontrivial homomorphism f and $C \in \{\mathrm{NC}, \mathrm{LT}, \mathrm{SLT}\}$.*

(1) *f preserves C.*

(2) *f preserves weakly C.*

(3) *f is injective and $f(\Sigma_1^*) \in C$.*

Proof. (1) $\Rightarrow$ (2) is obvious. (2) $\Rightarrow$ (3) follows from Lemma 4.1 and $\Sigma_1^* \in C$. (3) $\Rightarrow$ (1) follows from Proposition 4.2 and Corollary 4.1.

Note that Proposition 3.2 provides the following algorithm for checking injectivity of homomorphisms. A homomorphism f is injective iff for all $a \in \Sigma_1$, $f(a) \notin (f(\Sigma_1 - \{a\}))^*$ and $R \backslash R \cap R/R \cap \bar{R} = \varnothing$, where $R = f(\Sigma_1^*)$.

This algorithm and Theorem 4.2 provide algorithms for deciding whether a homomorphism preserves NC, LT, and SLT, respectively.

ACKNOWLEDGMENT

The authors thank the referee for referring them to several references and for many helpful suggestions.

References

1. R. McNaughton and S. Papert, "Counter-Free Automat," MIT Press, Cambridge, Mass., 1971.
2. Y. Zalcstein, Locally testable languages, *J. Comput. System Sci.* **6** (1972), 151–167.
3. M. P. Schützenberger, On finite monoids having only trivial subgroups, *Inform. Contr.* **8** (1965), 190–194.
4. J. A. Brzozowski, Derivatives of regular expressions, *J. Assoc. Comput. Mach.* **11** (1964), 481–494.
5. J. A. Brzozowski, Roots of star events, *J. Assoc. Comput. Mach.* **14** (1967), 466–477.
6. M. P. Schützenberger, On an application of some semigroup methods to some problems in coding, *IEEE Trans. Information Theory* **2** (1956), 47–60.
7. E. K. Blum, Free subsemigroups of a free semigroup, *Michigan Math. J.* **12** (1965), 179–182.
8. E. K. Blum, A note on free subsemigroups with two generators, *Bull. Amer. Math. Soc.* **71** (1965), 678–679.
9. J. A. Brzozowski and I. Simon, Characterizations of locally testable events, *Discrete Math.* **4** (1973), 243–271.
10. R. McNaughton, Algebraic decision procedures for local testability, *Math. Systems Theory* **8** (1974), 60–76.
11. A. Restivo, On a question of McNaughton and Papert, *Inform. Contr.* **25** (1974), 93–101.